# IT 230 Coding Activity Submission Template

Submit your work on the coding activities for Modules One, Two, Three, Four, and Six in this document. In addition to this document, you should submit a ZIP file containing all your Visual Studio project files and source code that can be run in Visual Studio on a different computer.

For each coding activity, complete the following steps:

* Download and rename this document to meet the file naming conventions requested in the assignment instructions.
* Fill in the required information below by replacing the bracketed text with the relevant information.
* Submit this document and your ZIP file for grading and feedback. Your ZIP file should follow the same naming conventions.

Document your work in the coding activity by completing each of the following items:

1. Provide a screenshot of the output that resulted from running your program successfully in Visual Studio. See the coding assignment instructions for an example of what should be included in the screenshot. Your screenshot must include the following elements:
   1. Your last name as the first printed text on the screen
   2. Verification that the program is fully functioning and data results are accurate for the given problem

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Copy and paste the source code text you wrote for this assignment from the \*.cs file into the space below. Only providing the \*.cs files or a screenshot does not meet the requirements for this part of the assignment. Code should be logically organized. It should also follow proper syntax and conventions noted in the Coding Activity Guidelines and Rubric.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WPFRegisterStudent

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Course choice;

private int totalCredit = 0; //Variable to keep track of the total credit hours of the user's selected classes

public MainWindow()

{

InitializeComponent();

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

Course course1 = new Course("IT 145");

Course course2 = new Course("IT 200");

Course course3 = new Course("IT 201");

Course course4 = new Course("IT 270");

Course course5 = new Course("IT 315");

Course course6 = new Course("IT 328");

Course course7 = new Course("IT 330");

this.comboBox.Items.Add(course1);

this.comboBox.Items.Add(course2);

this.comboBox.Items.Add(course3);

this.comboBox.Items.Add(course4);

this.comboBox.Items.Add(course5);

this.comboBox.Items.Add(course6);

this.comboBox.Items.Add(course7);

this.textBox.Text = "";

}

private void button\_Click(object sender, RoutedEventArgs e)

{

choice = (Course)(this.comboBox.SelectedItem);

if (this.listBox.Items.Contains(choice) && choice.IsRegisteredAlready())

{

this.label3.Content = "You have already registered for the " + choice.ToString() + " course"; //To properly inform the user they registered for the selected course already, I added string to label3

}

else if (totalCredit < 9) //Sets the maximum credit hours to 9, allowing the user to add a selected class if they are below 9 hours

{

this.listBox.Items.Add(choice); //Adds the course that the user selects to the listbox of courses

choice.SetToRegistered();

totalCredit += 3; //Since every course is 3 credit hours, when a course is added to the list the totalcredit goes up by 3

this.textBox.Text = Convert.ToString(totalCredit); //The credit hours are updated after selection

this.label3.Content = "Registration confirmed for course " + choice.ToString(); //Informs the user of the successful course registration

}

else

{

this.label3.Content = "You cannot register for more than 9 credit hours"; //Informs the user that they are at the total credit maximum, which means that they cannot sign up for another class

}

}

}

}

1. Show that you understand the task by explaining the design of your program in the space below. Include the process and steps you took to write your code. Explain how you arrived at the solution to the problem and completed the activity.

This is a program to help the user sign up for courses. It consists of a dropdown box with the list of course names and a button next to it to select to register for that course, a text box that keeps track of the courses the user is currently registered for, and a text box that keeps track of the total credit hours.   
I started by creating a private int variable called totalCredit for the total credit hours of the user. This variable was vital in order to keep track of the total credit hours by adding three credit hours every time a course was signed up for and to ensure that they can’t sign up for more courses once they reach 9 total credit hours. I needed to make a series of if-else statements next. The first if statement is used for when the user tries to register for a course that they are already signed up for and will inform the user of this error. An else if statement was needed next to set the maximum credit hours to 9 and confirm the registration of a class if the user is below 9 total credit hours. This will add the class to the course text list and add 3 to the total credit hours and inform the user of the successful registration. Finally, an else statement was needed for when the user has 9 total credit hours and tries to sign up for another course. When this happens, it informs the user that they are at the maximum total credit hours and cannot sign up for any more courses.

1. Reflect on your learning experience and what you learned from completing the activity.

This was the most complex assignment that we have completed yet in this class. I had a difficult time analyzing the different files and figuring out how it all worked together. After watching the video that shows how the program should work, I had a better understanding of what the missing code’s function was. This project made me realize just how helpful Visual Studio is since there were multiple occasions where I felt stuck, and its suggestions helped me figure out the next step I needed to take. I also realized some of my bad coding habits through this project such as forgetting to write notes right after writing the code or having the tendency to jump to fixing a part of a different function before finishing the previous one. This was an excellent experience for me and the most exciting assignment of this course for me since I was able to write more code rather than debugging!